FORSCHUNGSZENTRUM JÜLICH GmbH
Zentralinstitut für Angewandte Mathematik
D-52425 Jülich, Tel. (02461) 61-6402

Interner Bericht

Some Intricate Invariant Manifolds of Simple Dynamical Systems — a Maple Story

Edgar M. E. Wermuth

KFA-ZAM-IB-9518

August 1995
(Stand 03.08.95)

Submitted to The Maple Technical Newsletter

Some Intricate Invariant Manifolds of Simple Dynamical Systems — a Maple Story

Edgar M. E. Wermuth [1]

An Innocent-Looking Convergence Problem

In a recent analysis textbook the following problem is posed (Walter[12], p. 78):
Determine all α such that the sequence

$$x_0 = 1,\ x_1 = \alpha,\ x_{n+2} = x_{n+1}^2 - x_n \quad (n \geq 0) \tag{1}$$

is convergent.
It turns out to be not particularly easy to find a *single* α such that the sequence converges (see Walter[12], p. 365, for the sketch of a proof that there *exists* an α for which convergence holds)!

Nevertheless one can prove (see Wermuth[13]) that for every $x \geq 0$ there is exactly one $y > 0$ such that the sequence

$$x_0 = x,\ x_1 = y,\ x_{n+2} = x_{n+1}^2 - x_n \quad (n \geq 0) \tag{2}$$

is nonnegative and bounded and, in fact, converging monotonously to the limit 2. (The only possible limits of a sequence (1) or (2) are 2 and 0.) A stable numerical procedure to compute $y = f(x)$ is easily implemented in Maple. The simple idea: Start with the sequence

$$x, 2, 2, 2, 2, 2, 2, 2, \ldots$$

and replace every element except the first one by the square root of the sum of its two neighbours, working from left to right and always using the latest available value for each element; iterate this procedure (see Wermuth[13], Satz 4, for a convergence proof).

```
> func := proc(n:nonnegint,x)
>         local a,k,j;
>         a := array(0 .. n);
>         a[0] := 2.0 ;
>         for k to n do  a[k] := x od;
>         for k to n do  for j to k do  a[j] := sqrt(a[j-1]+a[j]) od od;
>         a[n]
>         end:
> func(20,1);
```

$$1.708875563$$

The procedure works not only for $x \geq 0$ but even for $x \geq -1.38$:

[1] Central Institute for Applied Mathematics, Research Centre Jülich (KFA), D-52425 Jülich, Germany.
E-mail: e.m.e.wermuth@kfa-juelich.de

```
> func(30,-1.38);
```

$$.2835163399$$

```
> func(50,-1.38);
```

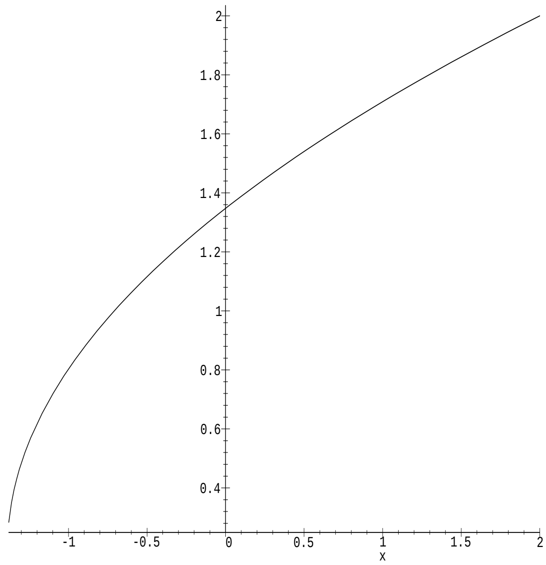
$$.2835076101$$

```
> ta:=time():plot('func(20,x)',x=-1.38..2):time_used:=time()-ta;
> ### use of ' very important here!!!
```

$$time\_used := .500$$

```
> plot('func(20,x)',x=-1.38..2,title=`Fig. 1`);
```

Fig. 1



To speed up the calculations we use hardware floats:

```
> ta:=time():plot('evalhf(func(20,x))',x=-1.37..2.01):time_used:=time()-ta;
> ### not significantly faster!
```

$$time\_used := .483$$

```
> fun:= proc(n,x) option remember;  evalhf(func(n,x)) end:
> ### remember computed values!!
```

```
> fun(13,0);
```

$$1.34755925807024535$$

```
> ta:=time():for i to 100 do func(20,47.11/i) od:time_used:=time()-ta;
```

$$time\_used := 17.867$$

```
> ta:=time():for i to 100 do fun(20,1147/i) od:time_used:=time()-ta;
```

$$time\_used := .633$$

Obviously 'plot' makes use of hardware floats!

A variant of 'fun' that rejects input < -1.38:

```
> fu := t -> if t >= -1 then evalhf(func(20,t)) elif t >= -1.38 then
> evalhf(func(30,t)) else ERROR('argument < -1.38') fi;

fu := proc(t)
      options operator,arrow;
          if -1 <= t then evalhf(func(20,t))
          elif -1.38  <= t then evalhf(func(30,t))
          else ERROR('argument < -1.38 ')
          fi
      end


> fu(-1.4);
Error, (in fu) argument < -1.38


> fu(1,2,3); ### extra arguments are ignored!!!
```

$$1.70887556327542955$$

```
> fun(44,1,fun);
```

$$1.70887556327542511$$

A procedure that computes $f : [a, b] \to \mathbf{R}$ as a list of $n + 1$ pairs:

```
> funpiece := proc(n:nonnegint,a:float,b:float)
>             local X,Y,j,x,y,u,v;
>             X := [seq(evalhf(a+j*(b-a)/n), j=0..n)];
>             Y := map(u->fun(20,u),X);
>             zip((x,y) -> [x,y],X,Y)
>             end:

> p0:=funpiece(1000,-1.38,2.):

> p0:=funpiece(1000,-1.38,2.):###second execution needs almost no time
```

It turns out that the graph of f is invariant under the transformation

$$\Phi : (x, y) \mapsto (x^2 - y, x), \tag{3}$$

and thus we can use it to extend the graph of f; the following procedure does the job:

```
> trafo := proc(n:nonnegint,a:list)
>          local j,x,b;
>          b:=a;
>          for j  to n do
>              b:=map(x -> [x[1]^2-x[2], x[1]], b);
>          od;
>          b;      ### this is only necessary in order to include n=0
>          end:
```

The result of applying the map (3) to the graph of f is quite surprising for someone not already familiar with the intricacies of dynamical systems:

```
> plot(trafo(3,p0),title=`Fig. 2`);
```

Fig. 2

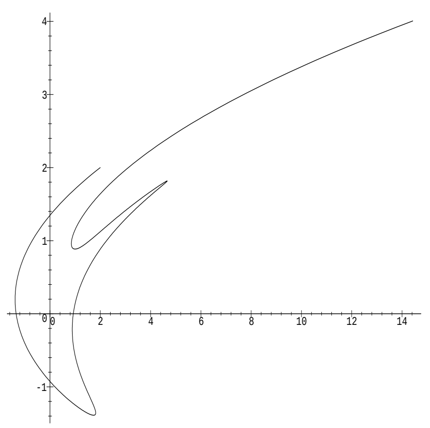
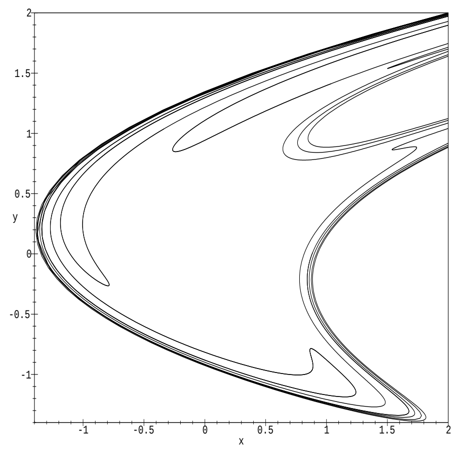


Fig. 5




```
> q0:=funpiece(2000,-1.38,1.):c4:=trafo(4,p0):c5:=trafo(5,q0):c6:=trafo(1,c5):
> plot({c4,c5,c6},title=`Fig. 3`):
```

Only a few steps more, and the resulting piece of curve will no longer fit into the known universe (whatever unit of length we assume)!

Now a closer look at the origin:

```
> plot({c4,c5,c6},x=-1.4..2,y=-1.4..2,title=`Fig. 4`):
```

(The reader should have Maple plot the figures not reproduced here – like Fig. 3 and Fig. 4 – on his local system. The numbering chosen refers to *all generated* figures.)

```
> r0:=funpiece(10000,-1.38,.7):c6a:=trafo(6,r0):c7:=trafo(1,c6a):c8:=trafo(1,c7):
> plot({c4,c5,c6,c6a,c7,c8},x=-1.4..2,y=-1.4..2,color=0,
> title=`Fig. 5`,axes=BOXED);
```

Doing this last plot already needs some computing power, although a completely smooth plot of c_8 would have required still more points. We used a DEC Alpha station with 48 Mbytes of main memory. (At this point, Maple memory has reached 37600 K, Maple CPU time 400 sec.)

Looking at Fig. 5 comes close to a complete solution of the convergence problem concerning the sequences (1) and (2):
It can be shown (see Wermuth[13]) that the whole curve a part of which is shown in Fig. 5 coincides with the set of all *points* (x, y) *in the plane such that the sequence* (2) *converges to* 2*. Thus the set of all* α *such that* (1) *converges to* 2 *is obtained by intersecting the curve with the line* $x = 1$*, a very complicated denumerable set! Details in Wermuth[14].*
What remains to be discussed is convergence to zero, and this turns out to be *impossible*! (See below and Wermuth[14])

Figures 1–5 show parts of the repelling (unstable) invariant manifold of the fixed point $(2, 2)$ of the map (3). According to an important theorem of Grobman and Hartman (see Hartman[6], chapter IX; Ruelle[10], Amann[1]) a nonlinear map behaves locally like a slight deformation of its linear approximation in the neighbourhood of a hyperbolic fixed point (i.e. a fixed point such that the Jacobian matrix has no eigenvalue of modulus one); the unstable and stable (attracting) invariant manifolds are tangent to the eigenspaces corresponding to eigenvalues of modulus > 1 and < 1, respectively. Thus the stable and unstable manifolds can be thought of as the nonlinear analogon or generalization of eigenspaces. In our example the attracting manifold is just the mirror image of the repelling one and thus is obtained from it by means of the following simple procedure.

(It's advisable to save the worksheet and to restart the Maple session at this point; the only previous material needed for the generation of the next three figures are the procedures 'func', 'fun', 'funpiece', and 'trafo'. Restarting the session from time to time is a remedy for Maple's greed for memory. Look up <?restart> for further information.)

```
> mirr:=proc(a:list)
>      local b,j,x:
>      b:= map(x->[x[2],x[1]],a)
> end:
```

We determine the points where both manifolds intersect; to this end we have to solve the equations $f(x) + x = x^2$ and $f(x) = x^2/2$.

```
> f:=x->fun(30,x):g1:=x->f(x)-x^2+x:fsolve(g1,-.8..-.5);
```

$$-.6366175408$$

```
> g2:=x->f(x)-x^2/2:fsolve(g2,-1.2..-1);
```

$$-1.154482684$$

```
> s0:=funpiece(1000,-.636617,2.):s1:=trafo(1,s0):t1:=mirr(s1):
> plot({s1,t1},color=0,title=`Fig. 6: The Pseudoseparatrix`);
```

Fig. 6: The Pseudoseparatrix

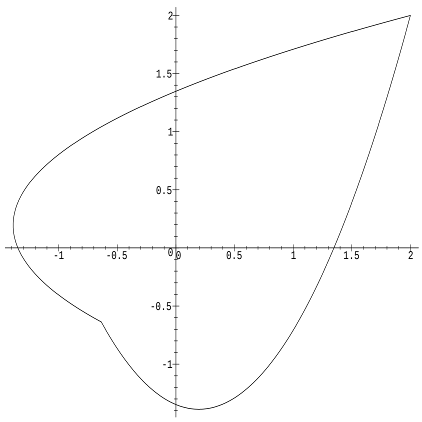
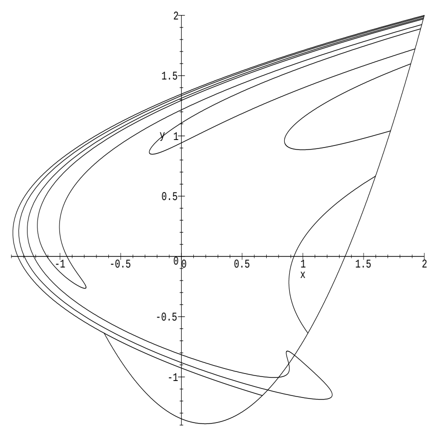


Fig. 7

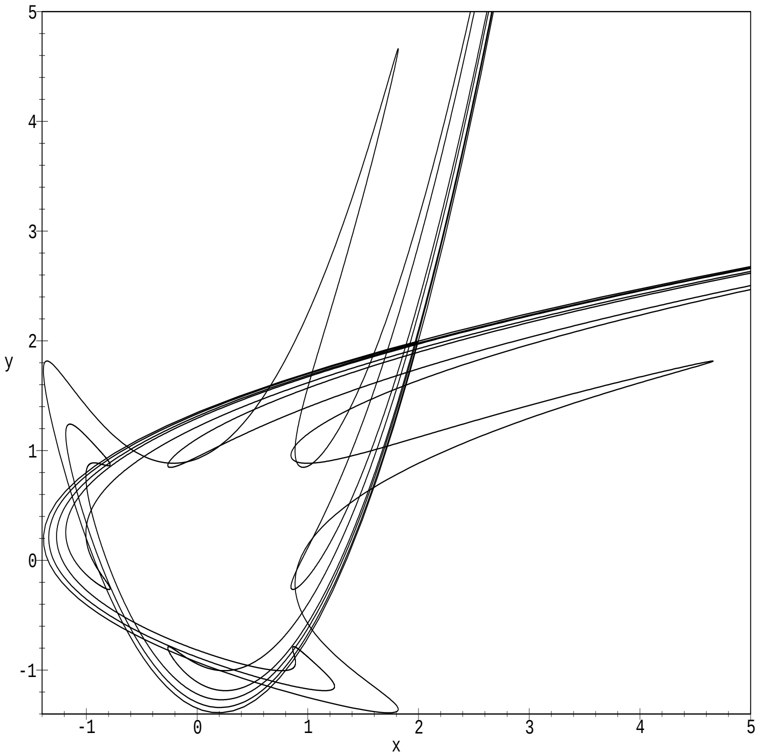



```
> pi0:=funpiece(1000,-1.15448,-.636617):pi1:=trafo(1,pi0):pi2:=trafo(1,pi1):
> pi3:=trafo(1,pi2):pi4:=trafo(1,pi3):pi5:=trafo(1,pi4):pi6:=trafo(1,pi5):
> plot({s1,t1,pi1,pi2,pi3,pi4,pi5,pi6},x=-1.4..2,y=-1.4..2,color=0,
> title=`Fig. 7`);
```

The pieces of the unstable manifold cut off by the stable manifold are successive images of each other, and all the enclosed areas are of equal size since the map (3) is area-preserving.

```
> p:=trafo(6,funpiece(15000,-1.15448,2.02)):q:=mirr(p):
> plot({p,q},x=-1.4..5,y=-1.4..5,color=0,
> title = `Fig. 8: The stable and unstable manifolds`,
> titlefont=[TIMES,ROMAN,16],axes=BOXED);
```

Fig. 8: The stable and unstable manifolds



Globally, these invariant manifolds have a very complicated structure. E.g., arbitrarily close to the fixed point there are points on the unstable manifold which intersect with the stable one (transversal homoclinic points); all Φ and Φ^{-1} images of these points belong to both manifolds. (See, e.g., Palis/Takens[9], chapter 2, for details.)

We now want to look at the neighbourhood of the other fixed point $(0, 0)$ of the map (3). This fixed point is *not* a hyperbolic one. The dynamical behaviour near $(0, 0)$, an elliptic fixed point (all eigenvalues have modulus one) of this area-preserving map, is *very* complicated. Studying some ad hoc invariant manifolds will lead to a first impression, but by no means to a complete picture. To construct invariant manifolds from pieces of curves we use 'trafo' and its inverse 'trafob'.

(Since we do not need anything but 'trafo' from the previous data and procedures, at this point again a restart is advisable.)

```
> trafob:=proc(n:nonnegint,a:list)
>        local j,x,b;
>        b:=a;
>        for j to n do
>            b:=map(x->[x[2],x[2]^2-x[1]],b);
>        od;
>        b;
>        end:
```

We define a curve with the end-points $(0,c)$ and (c,c^2). Extending it by means of 'trafo' and 'trafob' results in an invariant manifold of the map (3).

```
> curv:=(n,c)->[seq([k*c/n,evalhf(sqrt(c^2-(k*c/n)^2*(1-(k*c/n)^2)))],k=0..n)]:
Warning, `k` is implicitly declared local


> c0:=curv(200,.3):
> plot(c0,0..0.3,0..0.3,title=`Fig. 9`):
> ###x=0..0.3,y=0..0.3 would have implied labels
> plot({seq(trafo(k,c0),k=0..4)},color=0,title=`Fig. 10`):
> c20:=trafo(20,c0):c20b:=trafob(20,c0):
> plot({seq(trafo(k,c0),k=0..19),seq(trafo(k,c20),k=0..19)},
> title=`Fig. 11`,color=0):
> plot({seq(trafo(k,c0),k=0..19),seq(trafo(k,c20),k=0..19),seq(trafob(k,c0),
> k=1..19),seq(trafob(k,c20b),k=0..19)},title=`Fig. 12`,color=0):
> c25:=trafo(5,c20):c50:=trafo(25,c25):c75:=trafo(25,c50):
> c25b:=trafob(5,c20b):c50b:=trafob(25,c25b):c75b:=trafob(25,c50b):
> ta:=time(): ### A compromise between time and memory consumption !
>   plot({seq(trafo(k,c0),k=0..24),seq(trafo(k,c25),k=0..24),seq(trafo(k,c50),
>   k=0..24),seq(trafo(k,c75),k=0..24),seq(trafob(k,c0),k=1..24),
>   seq(trafob(k,c25b),k=0..24),seq(trafob(k,c50b),k=0..24),
>   seq(trafob(k,c75b),k=0..24)},title=`Fig. 13`,color=0,axes=BOXED);
>   time_used:= time()-ta;### This takes some time, but is worth the effort!
```

$$time\_used := 514.250$$

(On an IBM SP2 node this is done in less than 200 sec. On a smaller station like the 48Mbyte DEC Alpha one has to carefully avoid premature manipulation of the plot window – e.g. for saving a PostScript file – ; otherwise the process will be killed because of swap space problems. If possible, use a more powerful machine.)

The increasing symmetry of this invariant curve, the more complete it is, comes as a surprise; it is far from being obvious! (But after some reflection... See Wermuth[11])

```
> c96:=trafo(21,c75):c96b:=trafob(21,c75b):
> plot({seq(trafo(k,c96),k=0..3),seq(trafob(k,c96b),k=0..3)},title=`Fig. 14`,
> color=0):
> plot({seq(trafo(k,c96),k=0..3)},title=`Fig. 15`,color=0,axes=BOXED);
> c196:=trafo(100,c96):
> plot({seq(trafo(k,c196),k=0..3)},title=`Fig. 16`,color=0):
> c0h:=curv(1500,.3):c996h:=trafo(996,c0h):### This, too, needs time!
> plot({seq(trafo(k,c996h),k=0..3)},color=0,title=`Fig. 17`,axes=BOXED);
```

Observe that the last plot (Fig. 17) just like Fig. 15 shows only *four* successive trafo images of the initial piece of curve c_0 (Fig. 9)! By the way, Fig. 17 is a nice example to illustrate the Jordan curve theorem, though it is not *really* a closed curve.

Fig. 13

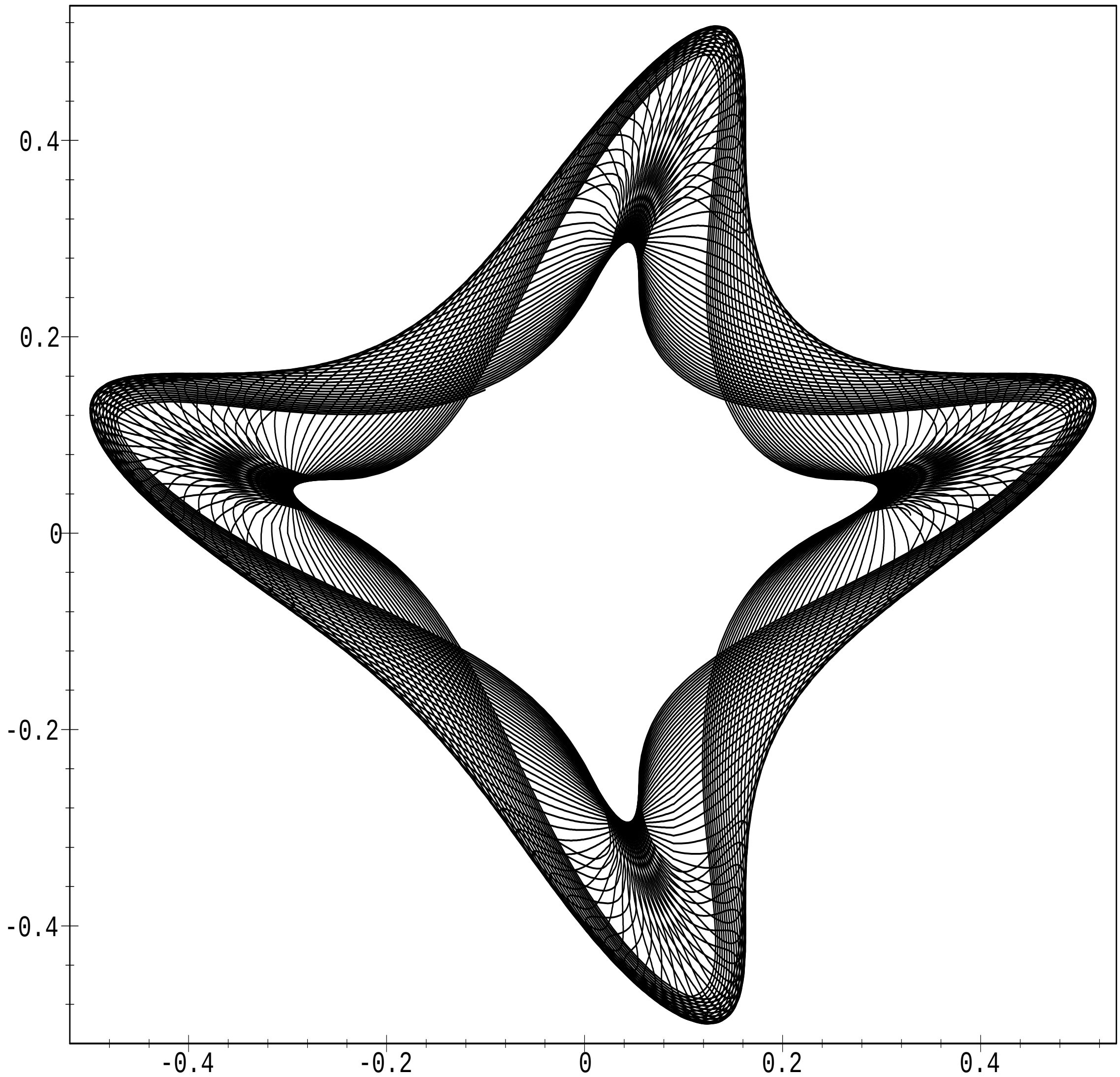


Fig. 15

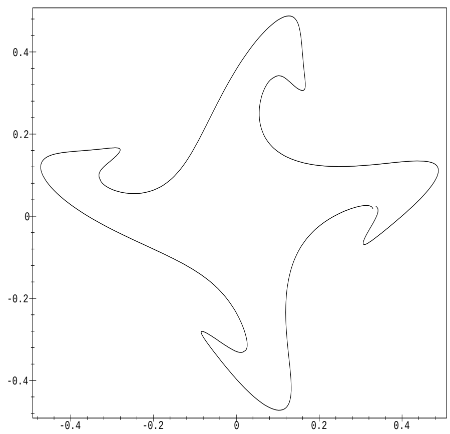
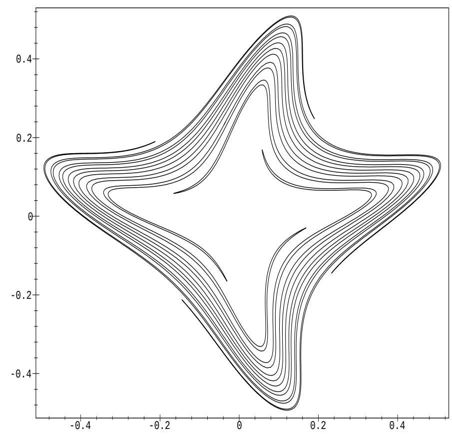


Fig. 17



Now we use a straight line as 'seed':

```
> lin:=(n,c)->[seq([evalhf(k*c/n),evalhf(c-k*(c-c^2)/n)],k=0..n)]:
Warning, `k` is implicitly declared local

> lin0:=lin(200,.3):plot({seq(trafo(k,lin0),k=0..4)},color=0,title=`Fig. 18`):
> plot({seq(trafo(k,lin0),k=0..9),seq(trafob(k,lin0),k=1..9)},
> title=`Fig. 19`,color=0):
> lin25:=trafo(25,lin0):lin50:=trafo(25,lin25):lin75:=trafo(25,lin50):
> lin25b:=trafob(25,lin0):lin50b:=trafob(25,lin25b):lin75b:=trafob(25,lin50b):
> plot({seq(trafo(k,lin0),k=0..24),seq(trafo(k,lin25),k=0..24),seq(trafo(k,
> lin50), k=0..24),seq(trafo(k,lin75),k=0..24),seq(trafob(k,lin0),k=1..24),
> seq(trafob(k,lin25b),k=0..24),seq(trafob(k,lin50b),k=0..24),
> seq(trafob(k,lin75b),k=0..24)},title=`Fig. 20`,color=0):
> lin396:=trafo(321,lin75):lin396b:=trafob(321,lin75b):
> plot({seq(trafo(k,lin396),k=0..3),seq(trafob(k,lin396b),k=0..3)},color=0,
> title=`Fig. 21`):
```

It starts quite differently, but in the end it's very much the same! The outer contours of Fig. 13 and Fig. 20 make visible the same invariant circle of Φ. (The *existence* of invariant circles in the case under consideration does not follow from Jürgen Moser's celebrated twist theorem (see, e.g., Wiggins[15], p. 151, and especially Siegel/Moser[11], §32ff.). Instead, the techniques developed in Arnold[2] are needed; see Wermuth[14] for details.)

Now we look at the behaviour closer to zero. Of course, the effects of the nonlinear term x^2 slow down considerably. But in the long run, this term exerts a strong influence on the dynamics. Watch!

```
> t0:=curv(200,.05):
> plot({seq(trafo(k,t0),k=0..9),seq(trafob(k,t0),k=1..9)},color=0,
> title=`Fig. 22`);
> t25:=trafo(25,t0):t50:=trafo(25,t25):t75:=trafo(25,t50):
> t25b:=trafob(25,t0):t50b:=trafob(25,t25b):t75b:=trafob(25,t50b):
```

```
> plot({seq(trafo(k,t0),k=0..24),seq(trafo(k,t25),k=0..24),seq(trafo(k,t50),
> k=0..24),seq(trafo(k,t75),k=0..24),seq(trafob(k,t0),k=1..24),seq(trafob(k,
> t25b), k=0..24),seq(trafob(k,t50b),k=0..24),seq(trafob(k,t75b),k=0..24)},
> title=`Fig. 23`,color=0);
```

Fig. 22

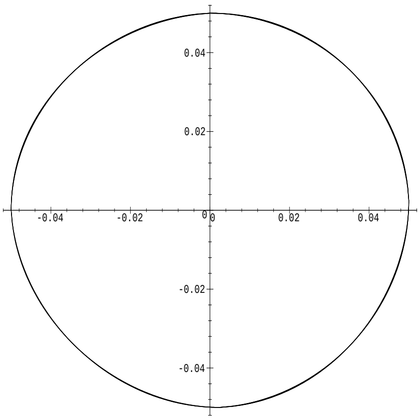
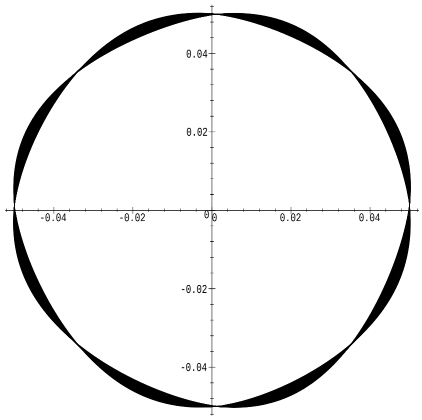


Fig. 23



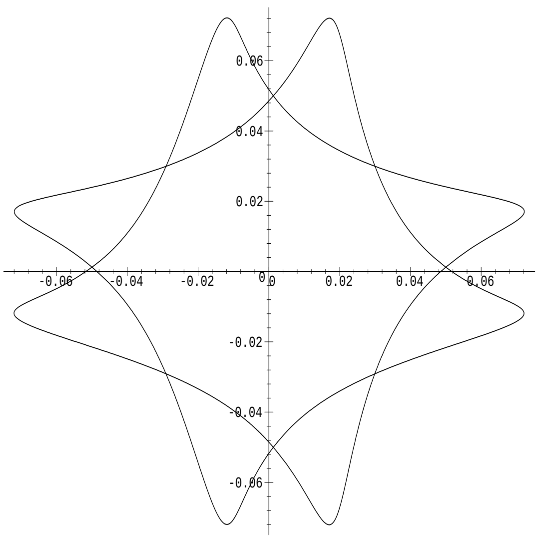

```
> t0h:=curv(500,.05):t996h:=trafo(996,t0h):t996bh:=trafob(996,t0h):

> plot({seq(trafo(k,t996h),k=0..3),seq(trafob(k,t996bh),k=0..3)},color=0,
> title=`Fig. 24`);

> t1996h:=trafo(1000,t996h):t1996bh:=trafob(1000,t996bh):

> plot({seq(trafo(k,t1996h),k=0..3),seq(trafob(k,t1996bh),k=0..3)},color=0,
> title=`Fig. 25`):
```

Fig. 24



Figs. 24 and 25 do *not* consist of exact invariant circles which are mirror images of each other (since Φ has no period 4 point).

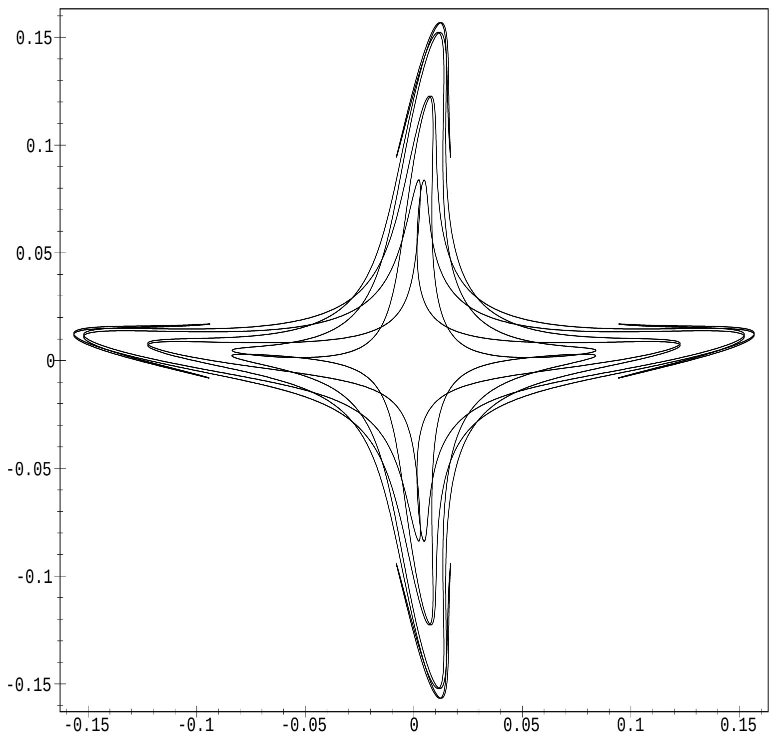
The next computation (an *overnight* job for my workstation) comes closer to an invariant structure.

```
> ta:=time():t0u:=curv(2000,.05):t9996u:=trafo(9996,t0u):t9996bu:=trafob(9996,
> t0u): time_used:=time()-ta; ### This will take some time!!!
```

$$time\_used := 29983.317$$

```
> plot({seq(trafo(k,t9996u),k=0..3),seq(trafob(k,t9996bu),k=0..3)},color=0,
> title=`Fig. 26: This took some time!`,titlefont=[TIMES,ROMAN,14],axes=BOXED);
```

Fig. 26: This took some time!



If we look at Φ and take into account that all intermediate computational results remain in the modulus range $(0.01, 0.2)$ we see that the rounding errors do not affect the reliability of the resulting picture.

We now generate invariant circles by simply plotting the iterates of a starting point.

```
> orbit:=proc(n:nonnegint,x,y)
>         local a ,k;
>         a:=vector(0..n);
>         a[0]:=[x,y];
>         for k to n do
```

```
>             a[k]:=[a[k-1][1]^2-a[k-1][2],a[k-1][1]]
>         od;
>         [seq(a[k],k=0..n)]
>         end:
```

Some experiments (the options 'style' and 'symbol' can also be chosen in the interactive plot menu):

```
> plot(orbit(2000,0,0.3),style=point,symbol=POINT,title=`Fig. 27`):
> plot(orbit(15000,0,0.3),style=point,symbol=POINT,title=`Fig. 28`):
> plot(orbit(15000,0,0.1),style=point,symbol=POINT,title=`Fig. 29`):
> plot(orbit(100,0,.5),style=point,symbol=POINT,title=`Fig. 30`):
> plot(orbit(300,0,.5),style=point,symbol=POINT,title=`Fig. 31`):
> plot(orbit(1000,0,.5),style=point,symbol=POINT,title=`Fig. 32`):
> plot(orbit(5000,0,.5),style=point,symbol=POINT,title=`Fig. 33`):
```

These experiments give strong evidence that all (or at least: many) points close to the origin belong to invariant circles, implying that convergence to zero is impossible for the sequences (1) and (2). The invariant circles have as symmetry axis the line $y = x$.

As it seems, the iterates of a point gradually spread densely on the invariant circle the original point belongs to. So there are two things to be proven:
- Every (?) point close to the origin belongs to a symmetric invariant circle;
- the iterates of a point on the circle are dense (but there are exceptions, since there are periodic points!).
See Wermuth[14] for details.

The most interesting region is the border that separates the invariant circles around $(0, 0)$ from the stable and unstable manifolds of the fixed point $(2, 2)$:

```
> plot({orbit(15000,0,.1),orbit(2000,0,.3),orbit(2000,0,.5),orbit(2000,0,.6),
> orbit(5000,0,.63),orbit(5000,0,.66),orbit(5000,0,.67),orbit(5000,0,.68),
> orbit(5000,0,.69)},color=0,style=point,symbol=POINT,axes=BOXED,
> title=`Fig. 34`); ### restart the session after this is done

> plot(orbit(5000,0,.69),style=point,symbol=POINT,title=`Fig. 35`);
```

Invariant circles around $(0, 0)$ and the inner frontier, a chain of 21 dusty islands.

Observe that the dynamics is more involved than the static picture of 'concentric' invariant circles around (0,0) suggests. Though Φ is an ever smaller deformation of a $\pi/2$ rotation near (0,0), points on different circles rotate quite differently, as is made visible, e.g., by Figs. 17 and 26.

There are also "second order islands":

```
> plot(orbit(20000,0,.685),style=point,symbol=POINT,title=`Fig. 36`):

> plot(orbit(20000,0,.685),0.48..0.56,0.48..0.56,style=point,symbol=POINT,
> title=`Fig. 37`):
```

A way of terminating the Maple session on my DEC Alpha station:

```
> plot({orbit(15000,0,.6855),,orbit(15000,0,.685),orbit(15000,0,.68),
> orbit(15000,0,.67)},0.48..0.56,0.48..0.56);
> ### This may terminate your Maple session!!
```

Fig. 34

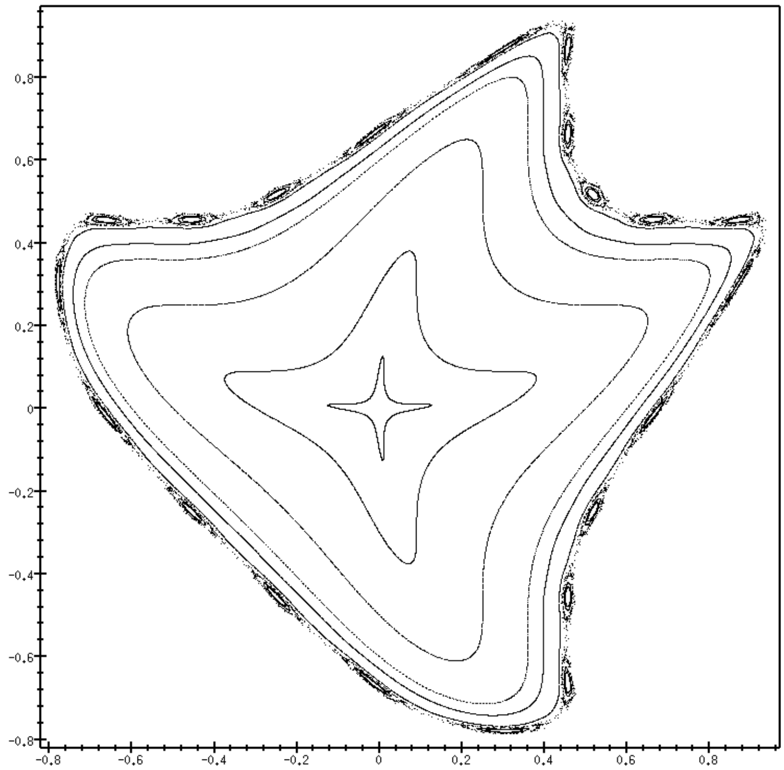


Fig. 35

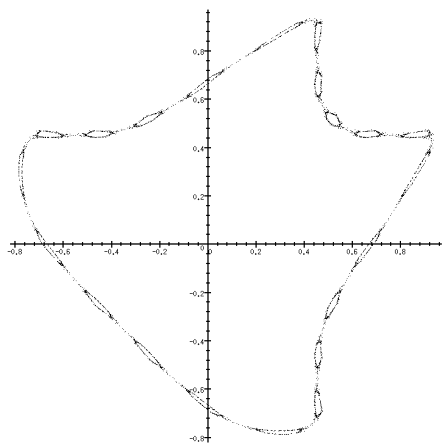
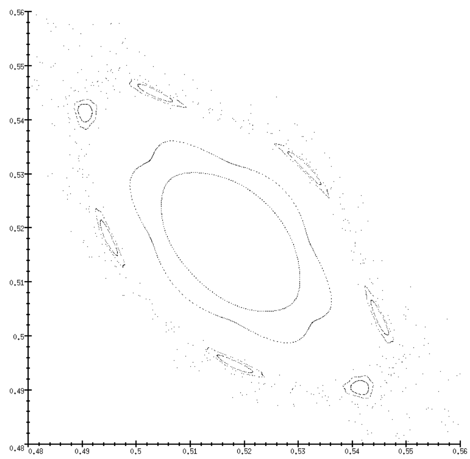


Fig. 36



A little less amount of data:

```
> p67:=orbit(10000,0,.67): p68:=orbit(10000,0,.68): p685:=orbit(10000,
> 0,.685): p6855:=orbit(10000,0,.6855): p69:=orbit(10000,0,.69):
```

```
> plot({p67, p68, p685, p6855, p69},0.48..0.56,0.48..0.56,
> style=point,symbol=POINT,title=`Fig. 38`); ### dust = p69
```

The first steps of an infinite hierarchy of elliptic fixed points of certain higher iterates of Φ (the map defined in (3)) and a corresponding hierarchy of "island chains" become visible.

Finally we generate an approximation of the complete picture (after restarting the session as explained earlier).

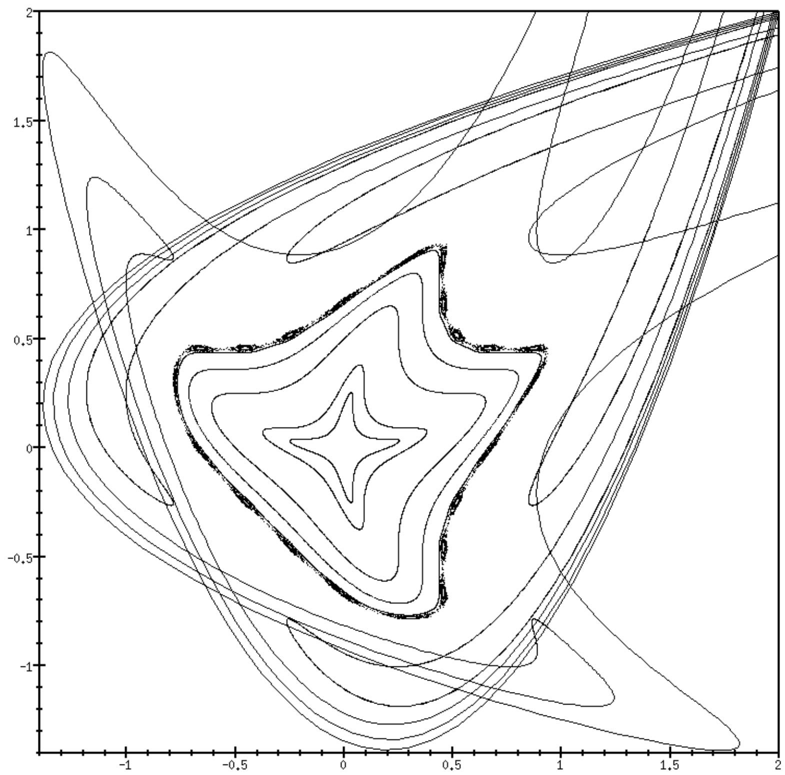
```
> inner:=plot({orbit(5000,0,.2),orbit(2000,0,.3),orbit(2000,0,.5),
> orbit(2000,0,.6),orbit(5000,0,.66),orbit(5000,0,.67),orbit(5000,0,.68),
> orbit(5000,0,.69)},color=0,style=point,symbol=POINT):
```

```
> d4:=mirr(c4):d5:=mirr(c5):d6:=mirr(c6):
```

```
> outer:=plot({c4,c5,c6,d4,d5,d6},-1.4..2,-1.4..2,color=0):
```

```
> with(plots):display({inner,outer},title=`Fig. 39`,axes=BOXED);
```

Fig. 39



This picture answers the convergence problem formulated at the outset!

(The quality of the last four reproduced figures does not equal that of the previous ones;

due to a minor bug in Maple V Release 3 concerning export as PostScript of plots with 'symbol=POINT' option these figures had to be generated from their worksheet representations. This bug will be fixed in Release 4.)

A Remarkable Analytic Function

We first define a procedure that approximately computes the real part of the analytic continuation $F(z)$ of the real function $f(x)$ for $x = \Re z \geq -1.38$. In principle, the iterative procedure 'func' from the first part also works in the complex case (see Wermuth[14]); but for reasons of efficiency the following (mathematically equivalent) approach via the procedures 'fucor' and 'fucoi' is to be preferred:

```
> func := proc(n:nonnegint,x)
>         local a,k,j;
>             a := array(0 .. n);
>             a[0] := 2.0 ;
>             for k to n do  a[k] := x od;
>             for k to n do  for j to k do  a[j] := sqrt(a[j-1]+a[j]) od od;
>             a[n]
>         end:
> func(20,-1+I);
```

$$1.028526737 + .5739317986\, I$$

```
> func(50,-1-I);
```

$$1.028526737 - .5739317986\, I$$

```
> fucor:=proc(n:integer,x,y)
>               local a, b, c, u, v, w, j, k,r:
>               a:=array(0..n);
>               b:=array(0..n);
>               a[0]:=2.0;
>               b[0]:=0;
>               c:=sqrt(.5);
>               for k to n do
>                     a[k]:=x;
>                     b[k]:=y
>               od;
>               for k to n do
>                     for j to k do
>                           u:=a[j-1]+a[j];
>                           v:=b[j-1]+b[j];
>                           w:=sqrt(sqrt(u*u+v*v)+u);
>                           a[j]:=c*w;
>                           b[j]:=c*v/w
>                     od
>               od;
>               a[n]
>               end:
```

(In the definition of fucor some care has been taken to avoid cancellation; e.g. the use of

$$\sqrt{u+iv} = \sqrt{\frac{1}{2}\sqrt{u^2+v^2}+u} + i\,\mathrm{sign}(v)\sqrt{\frac{1}{2}\sqrt{u^2+v^2}-u}$$

would have lead to an unstable iteration.)

```
> fucor(20,1,0);
```

$$1.708875563$$

```
> fucor(30,1,0);
```

$$1.708875563$$

To speed up the computations we use hardware floats:

```
> funcor:=(n,x,y)->evalhf(fucor(n,x,y)):
> ta:=time():funcor(50,3,3):time_used:=time()-ta;
```

$$time\_used := .100$$

```
> ta:=time():Re(func(50,2+2*I)):time_used:=time()-ta;
```

$$time\_used := 9.034$$

Now we define the imaginary part of F for $x = \Re z \geq -1.38$:

```
> fucoi:=proc(n:integer,x,y)
>               local a, b, c, u, v, w, k, j:
>               a:=array(0..n);
>               b:=array(0..n);
>               a[0]:=2.0;
>               b[0]:=0;
>               c:=evalhf(sqrt(.5));
>               for k to n do
>                     a[k]:=x;
>                     b[k]:=y
>               od;
>               for k to n do
>                     for j to k do
>                           u:=evalhf(a[j-1]+a[j]);
>                           v:=evalhf(b[j-1]+b[j]);
>                           w:=evalhf(sqrt(sqrt(u*u+v*v)+u));
>                           a[j]:=evalhf(c*w);
>                           b[j]:=evalhf(c*v/w)
>                     od
>               od;
>               b[n]
>               end:
> funcoi:=(n,x,y)->evalhf(fucoi(n,x,y)):
```

The definition of F:

```
> F:=z->funcor(20,Re(z),Im(z))+I*funcoi(20,Re(z),Im(z));
```

$$F := z \to \mathrm{funcor}(\,20, \Re(\,z\,), \Im(\,z\,)\,) + I\,\mathrm{funcoi}(\,20, \Re(\,z\,), \Im(\,z\,)\,)$$

```
> F(-1+I);
```

$$1.02852673685961338 + .573931798510020297\,I$$

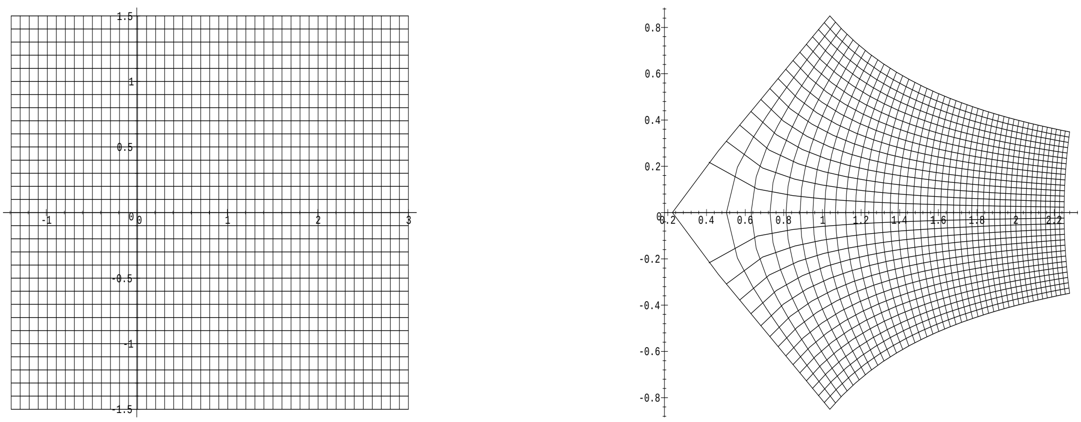
```
> with(plots):
> conformal(z,z=-1.388-1.5*I..3+1.5*I,grid=[45,31],
> title=`Fig. 40: A part of the z-plane`,color=white);
```

This grid is mapped by F onto the following one:

```
> conformal(F,-1.388-1.5*I..3+1.5*I,grid=[45,31],
> title=`Fig. 41: ... and its image under F`,color=white);
> ### x=-1.388 still works!
```

Fig. 40: A part of the z-plane

Fig. 41: ... and its image under F



The function F looks a little bit like a square root function (and $z = -1.388$ looks like a singularity!):

```
> conformal(sqrt(z+1.43),z=-1.388-1.5*I..3+1.5*I,grid=[45,31],title=`Fig. 42`):
```

Now we want to plot the real and imaginary parts of F as three-dimensional surfaces (in order to show that F looks *not* like a square root function!). The analytic continuation of F is obtained by means of the map (3), i. e. the transformation

$$(z, w) \mapsto (z^2 - w, z),$$

now applied to arbitrary complex z and w. The set of all pairs $(z, F(z))$ is an invariant manifold of this transformation in $\mathbf{C}^2$, and the graph of the analytic function of which F is a branch represents the totality of all pairs (z_0, z_1) such that the sequence $z_0, z_1, z_2, \ldots$ with $z_{n+2} = z_{n+1}^2 - z_n$ converges to 2 (Wermuth[14]).

```
> manif:=proc(n:nonnegint,x,y)
>             option remember;
>             if n=0 then [x,y,funcor(20,x,y),funcoi(20,x,y)]
>             else
>             [manif(n-1,x,y)[1]^2-manif(n-1,x,y)[2]^2-manif(n-1,x,y)[3],
>              2*manif(n-1,x,y)[1]*manif(n-1,x,y)[2]-manif(n-1,x,y)[4],
>              manif(n-1,x,y)[1],manif(n-1,x,y)[2]]
>             fi
>             end:
> manif(0,1,0);
```

$$[1, 0, 1.70887556327542955, 0]$$

```
> manif(5,1,0);
```

$$[1.037798606, 0, 1.412152923, 0]$$

```
> plot3d(['manif(0,x,y)[1]','manif(0,x,y)[2]','manif(0,x,y)[3]'],x=-1.38..2,
> y=-1..1,  title=`Fig. 43 R`,color=white);
> plot3d(['manif(0,x,y)[1]','manif(0,x,y)[2]','manif(0,x,y)[4]'],x=-1.38..2,
```

```
> y=-1..1,  title=‘Fig. 43 I‘,color=white);
```

We use the hidden line style, boxed axes, and a medium view projection.

Fig. 43 R

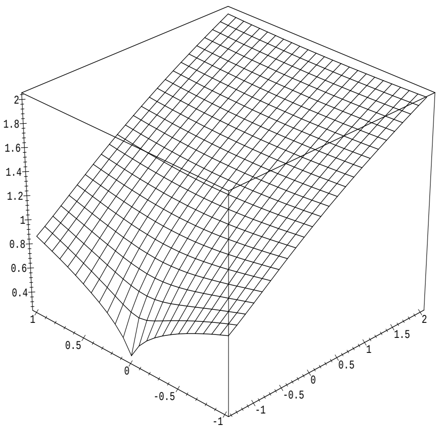
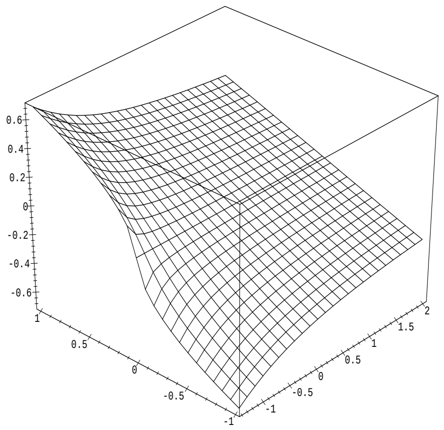


Fig. 43 I



The previous surfaces are parts of the correspondig following ones (F has a branch point at $z = -1.38...$):

```
> plot3d(['manif(1,x,y)[1]','manif(1,x,y)[2]','manif(1,x,y)[3]'],x=-1.38..2,
> y=-1..1, title=‘Fig. 44 R‘,color=white);
```

```
> plot3d(['manif(1,x,y)[1]','manif(1,x,y)[2]','manif(1,x,y)[4]'],x=-1.38..2,
> y=-1..1, title=‘Fig. 44 I‘,color=white);
```

Fig. 44 R

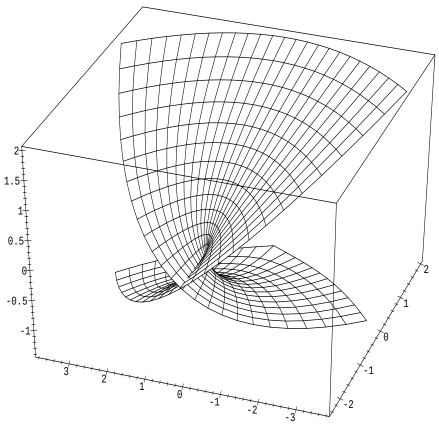
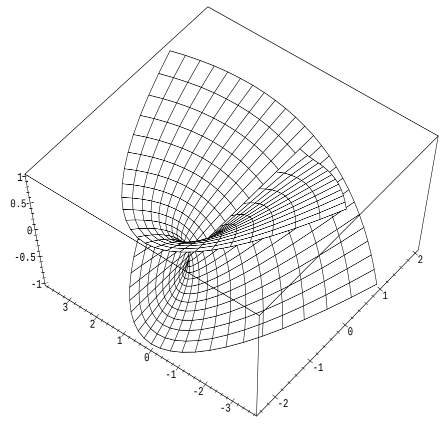


Fig. 44 I




```
> plot3d(['manif(2,x,y)[1]','manif(2,x,y)[2]','manif(2,x,y)[3]'],x=-1.38..2,
> y=-1..1,grid=[49,49], title=‘Fig. 45 R‘,color=white);
```

```
> plot3d(['manif(2,x,y)[1]','manif(2,x,y)[2]','manif(2,x,y)[4]'],x=-1.38..2,
> y=-1..1,grid=[49,49], title=‘Fig. 45 I‘,color=white);
```

We plot still larger portions of the real and imaginary part surfaces of F.

This time we have to look at the surfaces several times in order to understand them. The interactive rotation of the 3d plot is an invaluable help here.

Fig. 45 R

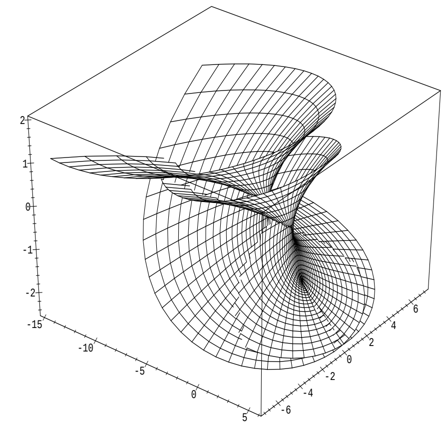
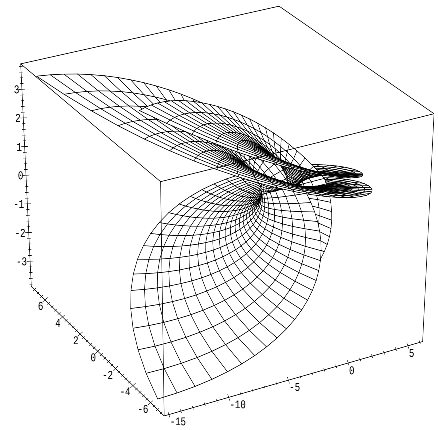


Fig. 45 I



We now take a closer look at the three branch points involved

```
> plot3d(['manif(2,x,y)[1]','manif(2,x,y)[2]','manif(2,x,y)[3]'],x=-1.38..2,
> y=-1..1,grid=[79,79],view=[-1.8..4,-1..1,-1.8..2],color=white,
> title=`Fig. 46 R: Three branch points`);
```

```
> plot3d(['manif(2,x,y)[1]','manif(2,x,y)[2]','manif(2,x,y)[3]'],x=-1.38..2,
> y=-1..1,grid=[79,79],view=[-1.8..4,0..1,-1.8..2],color=white,
> title=`Fig. 47 R: A median section`);
```

Fig. 46 R: Three branch points

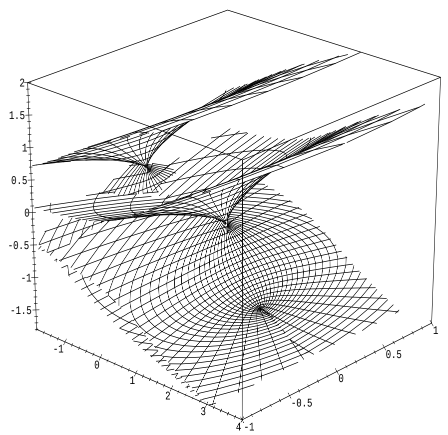
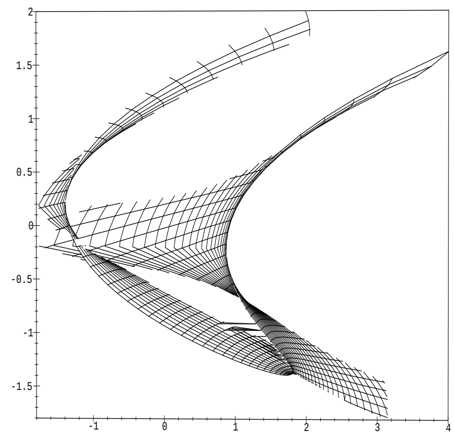


Fig. 47 R: A median section



This was but a glimpse at the formidable analytic function F. In fact, F has infinitely many branch points, all of them real and > -1.4, infinitely many in the interval $(-1.4, 1)$; on the other hand, the set of branch points is unbounded. (See Wermuth[14] for details.) F has a fixed point, $z = 2$, and we now compute the power series expansion of F about 2, having a radius of convergence > 3.38. For $F(z) = a_0 + a_1(z-2) + a_2(z-2)^2 + a_3(z-2)^3 + a_4(z-2)^4 + \ldots$ and its inverse $F^{-1}(z) = b_0 + b_1(z-2) + b_2(z-2)^2 + b_3(z-2)^3 + b_4(z-2)^4 + \ldots$ the following relations hold:

$$\begin{aligned} a_0 = b_0 = 2,\ a_1 = 2-\sqrt{3},\ b_1 = 2+\sqrt{3}, \\ a_2 = \frac{1}{2} - \frac{3}{10}\sqrt{3},\ b_2 = \frac{1}{2} + \frac{3}{10}\sqrt{3}, \\ a_k = -b_k \quad (k \geq 3). \end{aligned}$$

This leads to

$$a_k := \left(\mathrm{subs}\left(z=2, \mathrm{diff}\left(-b_2 \left(\sum_{j=1}^{k-1} a_j\,(z-2)^j\right)^2 + \left(\sum_{i=3}^{k-1} a_i \left(\sum_{j=1}^{k+1-i} a_j\,(z-2)^j\right)^i\right), z\,\$\,k\right)\right)\right) \Big/ \left(k!\,\left(b_1 - {a_1}^k\right)\right)$$

for $k \geq 3$ (see Wermuth[14]); fortunately Maple is willing to do this for us:

```
> Fcoeffs:= proc(n:nonnegint)
>                   local a,i,j,k,z;
>                   a:=array(0..max(3,n));
>                   a[0]:=2;
>                   a[1]:=2-sqrt(3);
>                   a[2]:=1/2-(3/10)*sqrt(3);
>                   for k from 3 to n do
>                         a[k]:=convert([seq(a[i]*(convert([seq(a[j]*(z-2)^j,
>                                   j=1..k+1-i)],`+`))^i,i=3..k-1)],`+`);
>                         a[k]:=-(1/2+(3/10)*sqrt(3))*(convert([seq(a[j]*(z-2)^j,
>                                   j=1..k-1)],`+`))^2+a[k];
>                         a[k]:=expand(radsimp(expand(subs(z=2,diff(a[k],z$k)))/
>                                   expand((k!)*(2+sqrt(3)-(2-sqrt(3))^k)),
>                                    'ratdenom')) ### to keep things simple
>                   od;
>                   [seq(a[k],k=0..n)];
>                   end:
```

(Using 'radnormal' instead of 'expand' would be about four times slower here.)

```
> Fcoeffs(6);
```

$$\left[2, 2-\sqrt{3}, \frac{1}{2} - \frac{3}{10}\sqrt{3}, \frac{1}{600}\sqrt{3}, -\frac{7}{22800}\sqrt{3}, \frac{289}{4560000}\sqrt{3}, -\frac{9079}{647520000}\sqrt{3}\right]$$

```
> evalf(");
```

$$[2., .267949192, -.0196152424, .002886751347, -.0005317699850, .0001097725183, -.00002428541092]$$

```
> [Fcoeffs(9)[9],Fcoeffs(9)[10]];
```

$$\left[-\frac{3556417493}{4564368480000000}\sqrt{3}, \frac{5263798543219}{274738467548160000000}\sqrt{3}\right]$$

```
> evalf(");
```

$$[-.1349561460\,10^{-5}, .3318489253\,10^{-6}]$$

We finally define the Taylor expansion operator FT, expanding F about $z = 2$:

```
> FT:=n->unapply( convert([seq(Fcoeffs(n)[k]*(z-2)^(k-1),k=1..n+1)],`+`),z):
Warning, `k` is implicitly declared local
```

```
> FT(7);
```

$$z \to 2 + \left(2 - \sqrt{3}\right)(z-2) + \left(\frac{1}{2} - \frac{3}{10}\sqrt{3}\right)(z-2)^2 + \frac{1}{600}\sqrt{3}(z-2)^3 - \frac{7}{22800}\sqrt{3}(z-2)^4 + \frac{289}{4560000}\sqrt{3}(z-2)^5 - \frac{9079}{647520000}\sqrt{3}(z-2)^6 + \frac{9330211}{2870672000000}\sqrt{3}(z-2)^7$$

```
> FT(11)(1);
```

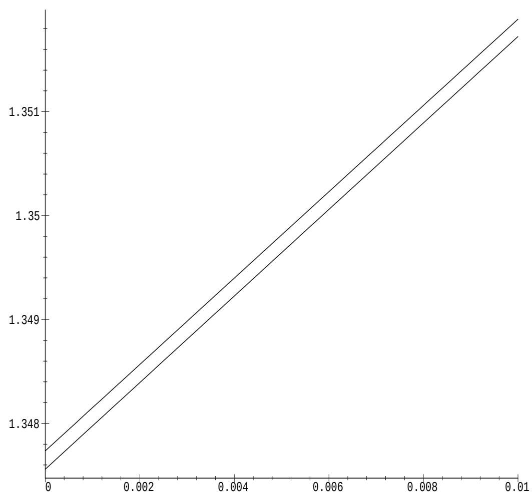
$$\frac{1}{2} + \frac{972446982326571172211175814 3}{1393301032699640064000000000 0}\sqrt{3}$$

```
> evalf("-F(1));
```

$$.8\,10^{-8}$$

```
> plot({F,FT(9)},0..0.01,title=`Fig. 48: A remarkable coincidence`,color=0);
```

Fig. 48: A remarkable coincidence



What else?

It is possible to treat more general recursions than (2) along the same lines. E.g., the reader may try his hands at

$$x_0 = x,\ x_1 = y,\ x_{n+2} = x_{n+1}^a - x_n \quad (n \geq 0)$$

for $a = 3, 4, 5, \ldots$ or even

$$x_0 = x,\ x_1 = y,\ x_{n+2} = \tan(x_{n+1}) - x_n \quad (n \geq 0).$$

See Wermuth[14].

Using Matlab

Of course, some of the numerical computations can be performed much faster and are less memory-consuming if we use a strictly numerically oriented system like Matlab. This is due to Matlab's simpler data structures and more powerful array-referencing possibilities. E.g., the 'trafo' procedure can be implemented in Matlab(!) as follows:

```
> function P = trafo(n,p)
> %
> %transforms n times an m by 2 matrix p
> %of m coordinate pairs of points in the plane;
> %the transformation used is (x,y)->(x^2-y,x)
> %
> P=p;
> for k=1:n
> q=P(:,1);
> P(:,1)=P(:,1).^2-P(:,2);
> P(:,2)=q;
> end;
```

This performs more than 200 times faster than the 'trafo' procedure we implemented in Maple! On the other hand, all computations *can* be done in Maple if we use a reasonably powerful workstation, the benefit being that the convenient Maple worksheet environment need not be left.

As an example, we show how to use Matlab in order to perform the transformations needed to generate Fig. 26. To facilitate the data exchange between Maple and Matlab it's advisable to start a Matlab session running in its own window from our Matlab directory, a *line mode* Maple session running in its own window from our Maple directory, and the Maple worksheet from our current Maple subdirectory. First we export the piece of curve t_{0u} (a list of lists in Maple) in flat ascii format from Maple to our Matlab directory; we use the line mode Maple session for this purpose, since the interface variable 'quiet' cannot be assigned the value 'true' in the workshit environment.

```
> curv:=(n,c)->[seq([k*c/n,evalhf(sqrt(c\

> ^2-(k*c/n)^2*(1-(k*c/n)^2)))],k=0..n)]:
Warning, 'k' is implicitly declared local
```

```
> t0u:=curv(2000,.05):
> interface(quiet=true);
> writeto(`../matlab/t0u.asc`);
> for k to 2001 do
> printf(`%g  %g\n`,t0u[k][1],t0u[k][2])
> od;
> writeto(terminal);
> interface(quiet=false);
```

Now we import the ascii file 't0u.asc' to the Matlab session by means of the Matlab command

```
> load t0u.asc
```

and use the Matlab routines 'trafo' and 'trafob' to transform the resulting 2001 by 2 matrix 't0u' (which, if necessary, can be easily reshaped in Matlab):

```
> t9996u = trafo(9996,t0u);
> t9996bu = trafob(9996,t0u);
```

(';' at the end of a Matlab command effects the same as ':' at the end of a Maple command.) The results are exported from Matlab to our current Maple subdirectory as raw ascii files by means of:

```
> save ../maple/dynamrekurs/MTech...
> /t9996u.asc t9996u -ascii
> save ../maple/dynamrekurs/MTech...
> /t9996bu.asc t9996bu -ascii
```

('...' is the Matlab equivalent of '\' at the end of an input line.) Finally we import the resulting ascii files back to our Maple worksheet:

```
> readlib(readdata):
> t9996u:=readdata(`t9996u.asc`,2):
> t9996bu:=readdata(`t9996bu.asc`,2):
> plot({seq(trafo(k,t9996u),k=0..3),seq(trafob(k,t9996bu),k=0..3)},
> color=0, title=`Fig. 26a: This took almost no time!`,
> titlefont=[TIMES,ROMAN,16],axes=BOXED):
```

Thus we need to invent other overnight jobs for our workstation.

FTP

A Maple ms file ('worksheet') with all the *Maple input* used in this article (but nothing else) can be obtained via anonymous ftp from our ftp server **ftp.zam.kfa-juelich.de**; directory: **/pub/unix/math/maple/msfiles**, filename: **manifoldinput.ms**

Acknowledgement

It's a pleasure to thank my colleagues at the Central Institute for Applied Mathematics in Jülich for the support and advice I received during the preparation of this article; especially I wish to thank Rita Peters, who always succeeded in persuading the DEC workstation I used to do what me or Maple wanted it to, and Johannes Grotendorst, who gave some advice concerning Maple usage as well as LaTeX code generation.

References

[1] Herbert Amann: *Gewöhnliche Differentialgleichungen*; Berlin 1983

[2] Vladimir I. Arnold: *Geometrical Methods in the Theory of Ordinary Differential Equations, 2nd edition*; Berlin 1988

[3] Robert M. Corless: *Essential Maple. An Introduction for Scientific Programmers*; New York 1995

[4] Kenneth J. Falconer: *Fractal Geometry. Mathematical Foundations and Applications*; Chichester 1990

[5] Jack K. Hale, Hüseyin Koçak: *Dynamics and Bifurcations*; New York 1991

[6] Philip Hartman: *Ordinary Differential Equations*; New York 1964

[7] André Heck: *Introduction to Maple*; New York 1993

[8] Michael Monagan: *Programming in Maple, the Basics*; Zürich 199?

[9] Jacob Palis, Floris Takens: *Hyperbolicity and Sensitive Chaotic Dynamics at Homoclinic Bifurcations*; Cambridge 1993

[10] David Ruelle: *Elements of Differentiable Dynamics and Bifurcation Theory*; San Diego 1989

[11] Carl Ludwig Siegel, Jürgen K. Moser: *Lectures on Celestial Mechanics*; Berlin 1971 (Reprinted 1995)

[12] Wolfgang Walter: *Analysis I, 2. Auflage*; Berlin 1990

[13] Edgar M. E. Wermuth: Fledermäuse in Flaschen-Über die Rekursion $x_{n+2} = x_{n+1}^2 - x_n$; Teil I, submitted

[14] Edgar M. E. Wermuth: Analytischer Salat-Über die Rekursion $x_{n+2} = x_{n+1}^2 - x_n$; Teil II, in preparation

[15] Stephen Wiggins: *Introduction to Applied Nonlinear Dynamical Systems and Chaos*; New York 1990